Search Results

Search Results for: [scope <sentence> claim<AND>((patent <sentence> (analysis or portforlio), scope, breadth, claim, word, count))]
Found 78 of 102,695 searched. → Rerun within the Portal

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 78 short listing

Prev Page 1 2 3 4 Next Page

1 The constituent object parser: syntactic structure matching for 94% information retrieval
Douglas P. Metzler , Stephanie W. Haas
ACM Transactions on Information Systems (TOIS) July 1989
Volume 7 Issue 3

The Constituent Object Parser is a shallow syntactic parser designed to produce dependency tree representations of syntactic structure that can be used to specify the intended meanings of a sentence more precisely than can the key terms of the sentence alone. It is intended to improve the precision/recall performance of information retrieval and similar text processing applications by providing more powerful matching procedures. The dependency tree representation and the relationship betwee ...

2 Getting around the task-artifact cycle: how to make claims and design by scenario 91%
John M. Carroll , Mary Beth Rosson
ACM Transactions on Information Systems (TOIS) April 1992
Volume 10 Issue 2

We are developing an “action science” approach to

human-computer interaction (HCI), seeking to better integrate activities directed at understanding with those directed at design. The approach leverages development practices of current HCI with methods and concepts to support a shift toward using broad and explicit design rationale to reify where we are in a design process, why we are there, and to guide reasoning about where we might go from there. We represent a designed artif ...

3 Checking relational specifications with binary decision diagrams 90%

Craig A. Damon , Daniel Jackson , Somesh Jha
ACM SIGSOFT Software Engineering Notes , Proceedings of the 4th ACM SIGSOFT symposium on Foundations of software engineering October 1996
Volume 21 Issue 6

Checking a specification in a language based on sets and relations (such as Z) can be reduced to the problem of finding satisfying assignments, or models, of a relational formula. A new method for finding models using ordered binary decision diagrams (BDDs) is presented that appears to scale better than existing methods.Relational terms are replaced by matrices of boolean formulae. These formulae are then composed to give a boolean translation of the entire relational formula. Throughout, boolea ...

4 Isomorph-free model enumeration: a new method for checking relational specifications 90%

Daniel Jackson , Somesh Jha , Craig A. Damon
ACM Transactions on Programming Languages and Systems (TOPLAS) March 1998
Volume 20 Issue 2

Software specifications often involve data structures with huge numbers of value, and consequently they cannot be checked using standard state exploration or model-checking techniques. Data structures can be expressed with binary relations, and operations over such structures can be expressed as formulae involving relational variables. Checking properties such as preservation of an invariant thus reduces to determining the validity of a formula or, equivalently, finding a model (of the form ...

5 Relational queries over interpreted structures 90%

Michael Benedikt , Leonid Libkin
Journal of the ACM (JACM) July 2000
Volume 47 Issue 4

We rework parts of the classical relational theory when the underlying domain is a structure with some interpreted operations that can be used in queries. We identify parts of the classical theory that go through 'as before' when interpreted structure is present, parts that go through only for classes of nicely behaved structures, and parts that only arise in the interpreted case. The first category include a number of results on language equivalence and expressive power characterizations ...

6 The elicitation of system knowledge by picture probes 90%
P. Barnard , M. Wilson , A. MacLean
ACM SIGCHI Bulletin , Conference proceedings on Human factors in computing systems April 1986
Volume 17 Issue 4

A technique is described in which a user's knowledge of a software package is elicited by means of a series of photographs depicting the system in a variety of states. The resultant verbal protocols were codified and scored in relation to the way in which the system actually worked. In the illustrative study described, the probes were administered twice after 5 and 10 hrs of system experience with an office product (VisiOn*). The number of true claims elicited increased with experience but ...

7 Public policy: Stay informed: participate in public policy discussion 90%
Bob Ellis
ACM SIGGRAPH Computer Graphics May 2002
Volume 36 Issue 2

8 Session 6D: agent analysis and validation: Model checking multi-agent systems with MABLE 89%
Michael Wooldridge , Michael Fisher , Marc-Philippe Huget , Simon Parsons
Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2 July 2002

MABLE is a language for the design and automatic verification of multi-agent systems. MABLE is essentially a conventional imperative programming language, enriched by constructs from the agent-oriented programming paradigm. A MABLE system contains a number of agents, programmed using the MABLE imperative programming language. Agents in MABLE have a mental state consisting of beliefs, desires and intentions. Agents communicate using request and inform performatives, in the style of the fipa agent ...

9 Software reliability and dependability: a roadmap 89%
Bev Littlewood , Lorenzo Strigini
Proceedings of the conference on The future of Software engineering May 2000

10 Automatic derivation of microsentences 89%
Basil T. Carmody , Paul E. Jones
Communications of the ACM June 1966
Volume 9 Issue 6

11 Towards a more complete model of role 89%
Cheh Goh , Adrian Baldwin
Proceedings of the third ACM workshop on Role-based access control October 1998

12 Critical slicing for software fault localization 88%
Richard A. DeMillo , Hsin Pan , Eugene H. Spafford
ACM SIGSOFT Software Engineering Notes , Proceedings of the 1996 international symposium on Software testing and analysis May 1996
Volume 21 Issue 3

Developing effective debugging strategies to guarantee the reliability of software is important. By analyzing the debugging process used by experienced programmers, we have found that four distinct tasks are consistently performed: (1) determining statements involved in program failures, (2) selecting suspicious statements that might contain faults, (3) making hypotheses about suspicious faults (variables and locations), and (4) restoring program state to a specific statement for verification. T ...

13 Hypertext, full text, and automatic linking 88%
J. H. Coombs
Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval December 1989

Current computing systems typically support only mid-century information structures: simple hierarchies. Hypertext technologies enable users to impose many structures on document sets and, consequently, provide many paths to desired information, but they require that users work their way through some structure. Full-text search eliminates this requirement by ignoring structure altogether. The search strategy can also be restricted to work within specified contexts.

The architecture provided ...

14 Transformations of CCP programs 88%

Sandro Etalle , Maurizio Gabbrielli , Maria Chiara Meo
ACM Transactions on Programming Languages and Systems (TOPLAS) May 2001
Volume 23 Issue 3

We introduce a transformation system for concurrent constraint programming (CCP). We define suitable applicability conditions for the transformations that guarantee the input/output CCP semantics is also preserved when distinguishing deadlocked computations from successful ones and when considering intermediate results of (possibly) nonterminating computations.The system allows us to optimize CCP programs while preserving their intended meaning: In addition to the usual benefits for sequential d ...

15 Parametric temporal logic for “model measuring” 88%

Rajeev Alur , Kousha Etessami , Salvatore La Torre , Doron Peled
ACM Transactions on Computational Logic (TOCL) July 2001
Volume 2 Issue 3

We extend the standard model checking paradigm of linear temporal logic, LTL, to a “model measuring” paradigm where one can obtain more quantitative information beyond a “Yes/No” answer. For this purpose, we define a parametric temporal logic, PLTL, which allows statements such as “a request p is followed in at most x steps by a response q,” where x is a free variable. We ...

16 Query optimization in a memory-resident domain relational calculus database system 88%

Kyu-Young Whang , Ravi Krishnamurthy
ACM Transactions on Database Systems (TODS) March 1990
Volume 15 Issue 1

We present techniques for optimizing queries in memory-resident database systems. Optimization techniques in memory-resident database systems differ significantly from those in conventional disk-resident database systems. In this paper we address the following aspects of query optimization in such systems and present specific solutions for them: (1) a new approach to developing a CPU-intensive cost model; (2) new optimization strategies for main-memory query processing; (3) new insight into ...

17 Why don't more non-North-American papers get accepted to CHI? 87%
Ellen A. Isaacs , John C. Tang
ACM SIGCHI Bulletin January 1996
Volume 28 Issue 1

18 Generating exception structures for legal information serving 87%
R. Winkels , D. J. B. Bosscher , A. W. F. Boer , J. A. Breuker
Proceedings of the seventh international conference on Artificial intelligence and law June 1999
More and more legal information is available in electronic form, but traditional retrieval mechanisms are insufficient to answer questions and legal problems of most users. In the ESPRIT project CLIME we are building a “Legal Information Server” (LIS), that not only retrieves all relevant norms for a user's query, but also applies them, giving the normative consequences of the 'situation' presented in the query. Typically, these queries represent very general an ...

19 The management of end user computing 87%
John F. Rockart , Lauren S. Flannery
Communications of the ACM October 1983
Volume 26 Issue 10
End users can be classified into six distinct types. Each of them needs differentiated education, support, and control from the Information Systems function. To support a large number of their applications a new computing environment, “the third environment” must be developed by Information Systems (I/S) management. Close attention must also be paid by I/S management to the need to involve “functional support personnel” (end users in each functional area who ...

20 Trust online 87%
Batya Friedman , Peter H. Khan , Daniel C. Howe
Communications of the ACM December 2000
Volume 43 Issue 12